CLAIMS:

1. A method for manipulating an instruction flow in a pipeline of a processor, comprising the following steps:
- detecting a stimulus leading to a disruption of progress of an instruction through a pipeline;
- on detecting said stimulus, forcing an instruction A required for responding to said stimulus by said processor directly into a first intermediate pipeline stage, said intermediate stage becoming available as a result of said disruption,
characterized in that said stimulus is detected from an instruction type of an instruction B residing in a second intermediate stage of the pipeline.

2. A method according to claim 1, characterized in that said instruction A causes the processor to store a processor status on a stack.

3. A method according to claim 1, characterized in that said instruction A causes the processor to retrieve a processor status from a stack.

4. A method according to claim 1, characterized in that said instruction B is an interrupt call that has been inserted into said first intermediate pipeline stage by said insertion means.

5. A method according to claim 1, characterized in that said instruction B is a programmable instruction causing a pipeline flush.

6. A method according to claim 5, characterized in that instruction A causes the processor to store a return address on a stack.

7. A system for manipulating an instruction flow, comprising:
- a processor having a processing pipeline;
- detection means (142) for detecting a stimulus leading to a disruption of the progress of an instruction through said pipeline;

- insertion means (180), responsive to said detection means, for forcing an instruction A directly into a first intermediate pipeline stage (126), said stage becoming available as a result of said disruption,
characterized in that said stimulus is detectable from an instruction type of an instruction B residing in a second intermediate stage of the pipeline (142).

8. A system according to claim 7, characterized in that:
- said instruction B is an element of an instruction bundle (420) comprising a plurality of instructions;
- said pipeline comprises a plurality of execute stages (362) for executing the plurality of instructions of said instruction bundle (420) in a parallel fashion, and
- said detections means (142) precedes the plurality of execute stages.

9. A system according to claim 8, characterized in that said detection means (142) is arranged to evaluate a bit pattern (440) attached to said instruction bundle (420), said bit pattern (440) marking the presence of said instruction type amongst said plurality of instructions.

10. A system according to claim 8 or 9, characterized in that said instruction bundle (420) is a Very Long Instruction Word (VLIW) in a compressed form.

11. A system according to one of the claims 7-10, characterized in that the instruction A to be forced into a pipeline by said insertion means is present in the system in a hard-coded manner.

12. A system according to one of the claims 7-10, characterized in that the instruction A to be forced into a pipeline by said insertion means is stored in a data storage device (284).

13. A computer program product, comprising a code module for execution by the system of claim 9, characterized in that said code module comprises an instruction extended with a bit pattern, said bit pattern making said instruction recognizable to the detection means of one of said systems.